

What is claimed is:

1. A method for solving a system of N linear equations in N unknown variables, the method comprising:
 - 5 (a) storing an estimate value for each unknown variable;
 - (b) initialising each estimate value to a predetermined value;
 - (c) for each estimate value:
 - (i) determining whether a respective predetermined condition is satisfied; and
 - 10 (ii) updating the estimate if and only if the respective predetermined condition is satisfied; and
 - (d) repeating step (c) until each estimate value is sufficiently close to an accurate value of the respective unknown variable.
- 15 2. A method according to claim 1, wherein said updating comprises adding a scalar value d to the respective estimate value, or subtracting a scalar value d from the respective estimate value.
3. A method according to claim 2, wherein said scalar value d is updated in a
20 predetermined manner.
4. A method according to claim 3, wherein said scalar value d is updated when and only when step (c) updates no estimate values.
- 25 5. A method according to claim 4, wherein said updating divides d by a scalar update value.
6. A method according to claim 5, wherein the scalar update value is equal to a power of two.
- 30 7. A method according to claim 6, wherein the scalar update value is equal to two.

8. A method according to claim 1, wherein each of said estimate values is initialised to be equal to zero.
- 5 9. A method according to claim 1, wherein the respective predetermined condition for each respective estimate value does not involve the respective estimate value.
- 10 10. A method according to claim 2, wherein the method establishes a respective auxiliary value for each estimate value.
11. A method according to claim 10, wherein said auxiliary values form an auxiliary vector \mathbf{Q} .
- 15 12. A method according to claim 11, wherein said predetermined condition for each respective estimate value involves the respective auxiliary value.
13. A method according to claim 12, wherein a plurality of auxiliary values are associated with each estimate value.
- 20 14. A method according to claim 13, wherein the predetermined condition for a respective estimate value involves the minimum amongst the plurality auxiliary values.
- 25 15. A method according to claim 14, wherein the minimum value is compared with a threshold value.
16. A method according to claim 15, wherein the condition is satisfied if the minimum value is less than the threshold value.
- 30

17. A method according to claim 16, wherein the plurality of auxiliary values for a respective estimate value consist of a first auxiliary value, and second auxiliary value which is the negative of the first auxiliary value.
- 5 18. A method according to claim 17, wherein the threshold value for the n th unknown variable is the scalar value d multiplied by the coefficient of the n th unknown variable in the n th equation.
- 10 19. A method according to claim 18, wherein one of a plurality of updates is selected in the condition is satisfied.
20. A method according to claim 19, wherein the scalar value d is added to the respective estimate value if the condition is satisfied and minimum value is the first auxiliary value.
- 15 21. A method according to claim 19, wherein the scalar value d is subtracted from the respective estimate value if the condition is satisfied and minimum value is the second auxiliary value.
- 20 22. A method according to claim 20, wherein the first auxiliary value for the n th unknown variable is initially set to be equal to the negative of the right hand side of the n th equation.
- 25 23. A method according to claim 21, wherein the first auxiliary value for the n th unknown variable is initially set to be equal to the negative of the right hand side of the n th equation.
24. A method according to claim 19, wherein the respective first and second
30 auxiliary values are updated if the condition is satisfied.

25. A method according to claim 24, wherein the first and second auxiliary values associated with each estimate value are updated if the condition is satisfied.

26. A method according to claim 25, wherein if the predetermined condition is satisfied for the n th estimate value:

the first auxiliary value for the m th estimate value is updated by:

 multiplying the coefficient of the m th unknown variable in the n th equation by the scalar value d ; and

 adding the result of said multiplication to the first auxiliary value to create a new first estimate auxiliary value, or subtracting the result of said multiplication from the first auxiliary value to create the new first estimate auxiliary value; and

the second auxiliary value for the m th estimate value is updated to be equal to the negative of the new first auxiliary value.

27. A method according to claim 1, wherein each estimate value is represented as a fixed point binary word.

28. A method according to claim 1, wherein each estimate value is a floating point binary word.

29. A method according to claim 1, wherein each estimate value is a complex number.

30. A method according to claim 3, wherein the scalar value d is updated such that the algorithm updates the estimate values in a bitwise manner, beginning with the most significant bit.

31. A method according to claim 4, wherein step (d) is carried out until a predetermined condition is satisfied.

32. A method according to claim 31, wherein said predetermined condition is a maximum number of iterations without an update to the scalar value d .

5 33. A method according to claim 32, wherein said predetermined condition is a total execution time elapsed without an update to the scalar value d .

34. A method according to claim 1, wherein the accurate solution of the equations is known to lie between upper and lower bounds, and the algorithm seeks a solution between said upper and lower bounds.

10

35. A method according to claim 34, wherein said estimate values are initialised to a value which is within said upper and lower bounds.

15 36. A method according to claim 35, wherein said estimate values are initialised to a value positioned at the midpoint of said upper and lower bounds.

37. A computer apparatus for solving a system of N linear equations in N unknown variables, the apparatus comprising:

20 a program memory containing processor readable instructions; and
a processor for reading and executing the instructions contained in the program memory;

wherein said processor readable instructions comprise instructions controlling the processor to carry out the method according to claim 1.

25 38. A data carrier carrying computer readable program code to cause a computer to execute procedure in accordance with the method of claim 1.

39. A method for solving a system of N linear equations in N unknown variables, the method comprising:

- 30 (a) storing an estimate value for each unknown variable;
(b) initialising each estimate value to a predetermined value;
(c) attempting to update each estimate value using a scalar value d ;

- (d) updating the scalar value if no updates are made in step (c); and
- (e) repeating step (c) and step (d) until each estimate value is sufficiently close to an accurate value of the respective unknown variable.

5 40. A method according to claim 39, wherein updating said estimate values comprises adding the scalar value d to an estimate value, or subtracting the scalar value d from an estimate value

41. A method according to claim 40, wherein said updating the scalar value
10 divides the scalar value by a scalar update value.

42. A method according to claim 41, wherein the scalar update value is equal to a power of two.

15 43. A method according to claim 42, wherein the scalar update value is equal to two.

44. A method according to claim 39, wherein each of said estimate values is initialised to be equal to zero.

20

45. A method according to claim 39, wherein step (c) comprises:
for each estimate value:

- (i) determining whether a respective predetermined condition is satisfied; and
- (ii) updating the estimate if and only if the respective predetermined
25 condition is satisfied;

46. A method according to claim 45, wherein the method establishes a respective auxiliary value for each estimate value.

30 47. A method according to claim 46, wherein said auxiliary values form an auxiliary vector \mathbf{Q} .

48. A method according to claim 47, wherein said predetermined condition for each respective estimate value involves the respective auxiliary value.
49. A method according to claim 48, wherein a plurality of auxiliary values are associated with each estimate value.
50. A method according to claim 49, wherein the predetermined condition for a respective estimate value involves the minimum amongst the plurality auxiliary values.
51. A method according to claim 50, wherein the minimum value is compared with a threshold value.
52. A method according to claim 51, wherein the condition is satisfied if the minimum value is less than the threshold value.
53. A method according to claim 52, wherein the plurality of auxiliary values for a respective estimate value consist of a first auxiliary value, and second auxiliary value which is the negative of the first auxiliary value.
54. A method according to claim 53, wherein the threshold value for the n th unknown variable is the scalar value d multiplied by the coefficient of the n th unknown variable in the n th equation.
55. A method according to claim 54, wherein one of a plurality of updates is selected if the condition is satisfied.
56. A method according to claim 55, wherein the scalar value d is added to the respective estimate value if the condition is satisfied and minimum value is the first auxiliary value.

57. A method according to claim 56, wherein the scalar value d is subtracted from the respective estimate value if the condition is satisfied and minimum value is the second auxiliary value.
58. A method according to claim 56, wherein the first auxiliary value for the n th unknown variable is initially set to be equal to the negative of the right hand side of the n th equation.
59. A method according to claim 57, wherein the first auxiliary value for the n th unknown variable is initially set to be equal to the negative of the right hand side of the n th equation.
60. A method according to claim 55, wherein the respective first and second auxiliary values are updated if the condition is satisfied.
61. A method according to claim 60, wherein the first and second auxiliary values associated with each estimate value are updated if the condition is satisfied.
62. A method according to claim 61, wherein if the predetermined condition is satisfied for the n th estimate value:
the first auxiliary value for the m th estimate value is updated by:
 multiplying the coefficient of the m th unknown variable in the n th equation by the scalar value d ; and
 adding the result of said multiplication to the first auxiliary value to create a new first estimate auxiliary value, or subtracting the result of said multiplication from the first auxiliary value to create a new first estimate auxiliary value; and
the second auxiliary value for the m th estimate value is updated to be equal to the negative of the new first auxiliary value.

63. A method according to claim 39, wherein each estimate value is represented as a fixed point binary word.
- 5 64. A method according to claim 39, wherein each estimate value is a floating point binary word.
65. A method according to claim 39, wherein each estimate value is a complex number.
- 10 66. A method according to claim 39, wherein step (e) is carried out until a predetermined condition is satisfied.
67. A method according to claim 66, wherein said predetermined condition is a maximum number of iterations without an update to the scalar value d .
- 15 68. A method according to claim 66, wherein said predetermined condition is a total execution time elapsed without an update to the scalar value d .
69. A method according to claim 39, wherein the accurate solution of the equations is known to lie between upper and lower bounds, and the algorithm seeks a solution between said upper and lower bounds.
- 20 70. A method according to claim 39, wherein said estimate values are initialised to a value which is within said upper and lower bounds.
- 25 71. A method according to claim 70, wherein said estimate values are initialised to a value positioned at the midpoint of said upper and lower bounds.
72. A computer apparatus for solving a system of N linear equations in N unknown variables, the apparatus comprising:
- 30 a program memory containing processor readable instructions; and

a processor for reading and executing the instructions contained in the program memory;

wherein said processor readable instructions comprise instructions controlling the processor to carry out the method according to claim 39.

5

73. A data carrier carrying computer readable program code to cause a computer to execute procedure in accordance with the method of claim 39.

74. A method for solving a system of N linear equations in N unknown variables of the form:

10

$$\mathbf{R}\mathbf{h} = \boldsymbol{\beta}$$

the method comprising:

15 generating a quadratic function of the form:

$$J(\mathbf{h}) = \sum_{m=1}^N \sum_{n=1}^N \mathbf{R}(m,n) \mathbf{h}(m) \mathbf{h}(n) - 2 \sum_{n=1}^N \boldsymbol{\beta}(n) \mathbf{h}(n); \text{ and}$$

minimising said function using co-ordinate descent optimisation;

wherein \mathbf{R} is a coefficient matrix of the system of linear equations; \mathbf{h} is a vector of the N unknown variables; $\boldsymbol{\beta}$ is a vector containing the value of the right hand side of each equation; $\mathbf{R}(m,n)$ is an element of the matrix \mathbf{R} ; $\mathbf{h}(m)$ is the m th element of the matrix \mathbf{h} ; and $\boldsymbol{\beta}(n)$ is the n th element of the vector $\boldsymbol{\beta}$.

20

75. A computer processor configured to solve a system of N linear equations in N unknown variables, comprising:

25

storage means for storing an estimate value for each unknown variable;

storage means for storing coefficients of each unknown variable in each equation;

storage means for storing a scalar value d ;

initialising means for initialising each estimate value;

computing means configured to process each estimate value by determining whether a respective predetermined condition is satisfied, and to update the estimate if and only if the respective predetermined condition is satisfied, said computing means being configured to repeatedly process each estimate value until each estimate value is sufficiently close to an accurate value of the respective unknown variable.

76. A computer processor according to claim 75, further comprising update means for updating the scalar value d .

77. A computer processor according to claim 76, wherein said update means divides the value of the scalar value d by a value equal to a power of two.

78. A computer processor according to claim 77, wherein said update means divides the value of the scalar value d by a value equal to two.

79. A computer processor according to claim 77, wherein said update means is a bit shift device.

80. A computer processor configured to solve a system of N linear equations in N unknown variables, comprising:

storage means for storing an estimate value for each unknown variable;

storage means for storing coefficients of each unknown variable in each equation;

storage means for storing a scalar value d ;

initialising means for initialising each estimate value;

computing means configured to:

(a) attempt to update each estimate value using a scalar value d ,

(b) update the scalar value d if no updates are made in step (a);

and

(c) repeat step (a) and step (b) until each estimate value is sufficiently close to an accurate value of the respective unknown variable.

81. A multiuser receiver device for obtaining data transmitted by a plurality of users, the device comprising:

5 a plurality of filters, each filter being arranged to filter out a spreading code used by a respective user;

equation solving means to find a solution \mathbf{h} of a system of linear equations of the form $\mathbf{R}\mathbf{h} = \beta$ where \mathbf{R} is the cross correlation of the spreading codes used by the plurality of users, and β is a vector containing the filter output signals; and

10 means to obtain the transmitted data using a solution provided by the equation solving means;

wherein the equation solving means:

(a) stores an estimate value for each value of the solution \mathbf{h} ;

(b) initialises each estimate value to a predetermined value;

(c) for each estimate value:

15 (i) determines whether a respective predetermined condition is satisfied; and

(ii) updates the estimate if and only if the respective predetermined condition is satisfied; and

(d) repeats step (c) until each estimate value is sufficiently close to an
20 accurate value of the respective unknown variable.

82. A multiuser receiver device for obtaining data transmitted by a plurality of users , the device comprising:

25 a plurality of filters, each filter being arranged to filter out a spreading code used by a respective user;

equation solving means to find a solution \mathbf{h} of a system of linear equations of the form $\mathbf{R}\mathbf{h} = \beta$ where \mathbf{R} is the cross correlation of the spreading codes used by the plurality of users, and β is a vector containing the filter output signals; and

30 means to obtain the transmitted data using a solution provided by the equation solving means;

wherein the equation solving means:

- (a) stores an estimate value for each unknown variable;
- (b) initialises each estimate value to a predetermined value;
- (c) attempts to update each estimate value using a scalar value d ;
- (d) updates the scalar value if no updates are made in step (c); and
- 5 (e) repeats step (c) and step (d) until each estimate value is sufficiently close to an accurate value of the respective unknown variable.

83. A method for generating filter coefficients for use in an echo cancellation apparatus, the method comprising:

- 10 (a) generating a cross correlation matrix \mathbf{R} containing the cross correlation of first and second signals and ;
- (b) generating an auto correlation vector β containing an autocorrelation of the first signal; and
- (c) determining a vector \mathbf{h} for which $\mathbf{R}\mathbf{h} = \beta$, said vector \mathbf{h} containing
- 15 the said filter coefficients;

wherein the vector \mathbf{h} is determined by solving the system of equations $\mathbf{R}\mathbf{h} = \beta$ by:

- (d) storing an estimate value for each element of the vector \mathbf{h} ;
- (e) initialising each estimate value to a predetermined value;
- (f) for each estimate value:
- 20 (i) determining whether a respective predetermined condition is satisfied; and
- (ii) updating the estimate if and only if the respective predetermined condition is satisfied; and
- (g) repeating step (f) until each estimate value is sufficiently close to an
- 25 accurate value of the respective unknown variable.

84. A method for generating filter coefficients for use in an echo cancellation apparatus, the method comprising:

- 30 (a) generating a cross correlation vector β containing the cross correlation of first and second signals;

- (b) generating an auto correlation matrix \mathbf{R} containing an autocorrelation of the first signal; and
 - (c) determining a vector \mathbf{h} for which $\mathbf{R}\mathbf{h} = \beta$, said vector \mathbf{h} containing the said filter coefficients;
- 5 wherein the vector \mathbf{h} is determined by solving the system of equations $\mathbf{R}\mathbf{h} = \beta$ by:
- (d) storing an estimate value for each unknown variable;
 - (e) initialising each estimate value to a predetermined value;
 - (f) attempting to update each estimate value using a scalar value d ;
 - (g) updating the scalar value if no updates are made in step (f); and
 - 10 (h) repeating step (g) and step (h) until each estimate value is sufficiently close to an accurate value of the respective unknown variable.
85. A method for solving a system of linear equations, comprising the steps of:
- a. representing elements of a solution vector as fixed point binary words
 - 15 each consisting of at least one bit;
 - b. initialising the solution vector and an auxiliary vector;
 - c. performing, for each bit representing the binary words, bit-wise iterations comprising the steps of:
 - i. performing passes through all elements of the solution vector;
 - 20 ii. updating elements of the solution vector in the passes;
 - iii. updating elements of the auxiliary vector in the passes;
 - iv. repeating the passes until a finishing condition is fulfilled;
 - d. stopping solving the system of linear equations when a stopping condition is fulfilled.
- 25 86. The method as defined in claim 85, wherein elements of the solution vector are initialised as zeros.
87. The method as defined in claim 85, wherein the auxiliary vector is initialised by the right-side vector of the system of linear equations.

30

88. The method as defined in claim 85, wherein the bit-wise iterations start from the most significant bit and proceed with the next less significant bit if the finishing condition is fulfilled.
- 5 89. The method as defined in claim 85, wherein in each pass, in turn, for each element of the solution vector a condition successful/unsuccessful is checked.
90. The method as defined in claim 85, wherein the finishing condition is fulfilled if in a pass no element of the solution vector is updated.
- 10 91. The method as defined in claim 85, wherein the stopping condition is fulfilled if a predefined number of passes through all elements of the solution vector is exceeded.
- 15 92. The method as defined in claim 85, wherein the stopping condition is fulfilled if a predefined number of bit-wise iterations, defining the number of valuable bits in elements of the solution vector as well as accuracy of the solution, is exceeded.
- 20 93. The method as defined in claim 85, wherein the stopping condition is fulfilled if a computer time predefined for performing this method is finished.
94. The method as defined in claim 85, wherein when passing through elements of the solution vector the order of analysing elements of the solution vector in the pass is arbitrary.
- 25 95. The method as defined in claim 85, wherein when passing through elements of the solution vector the pass starts from an element whose position corresponds to the position of an element of the auxiliary vector with maximum amplitude and in the order of reducing the amplitude.
- 30 96. The method as defined in claim 95, wherein ordering elements of the auxiliary vector is performed to define the order of elements in the pass.

97. The method as defined in claim 89 wherein updating elements of the solution vector and the auxiliary vector is performed only if the condition successful/unsuccessful is successful.
- 5
98. The method as defined in claim 97, wherein the only element of the solution vector is updated, for which the condition successful/unsuccessful is checked.
99. The method as defined in claim 98, wherein a finite number of possible updates of the element of the solution vector are analysed for finding a preferable update and the element of the solution vector when updated is set to be equal to the preferable update.
- 10
100. The method as defined in claim 99 wherein finding the preferable update comprises the steps of:
- 15
- e. calculating, for each possible update, an auxiliary value;
 - f. finding a minimum among the auxiliary values;
 - g. calculating a threshold;
 - h. comparing the minimum with the threshold;
 - 20 i. choosing the preferable update as that corresponding to the minimum.
101. The method as defined in claim 100, wherein the condition successful/unsuccessful is successful if the minimum among the auxiliary values is less than the threshold, and the condition successful/unsuccessful is unsuccessful if
- 25 the minimum among the auxiliary values is higher than or equal to the threshold.
102. The method as defined in claim 100, wherein calculating the auxiliary values is based on the corresponding element of the auxiliary vector.
- 30 103. The method as defined in claim 100, wherein calculating the threshold is performed by using a diagonal element of the coefficient matrix, the diagonal

element corresponding to the element of the solution vector, and a step-size parameter.

104. The method as defined in claim 103, wherein the step-size parameter is
5 decreased after each bit-wise iteration.

105. The method as defined in claim 104, wherein decreasing the step-size is by a factor of two.

10 106. The method as defined in claim 95, wherein elements of the auxiliary vector are updated by using elements of the coefficient matrix and the step-size parameter.

107. The method as defined in claim 106, wherein an element of the auxiliary vector is updated by using elements of a row of the coefficient matrix, the row
15 corresponding to the updated element of the auxiliary vector, and the step-size parameter.

108. A computer system for solving a system of linear equations, comprising:
j. a host processor producing itself or receiving from other devices
20 parameter signals representing elements of a coefficient matrix and right-side vector of the system of linear equations and transmitting to other devices parameter signals representing elements of a solution vector;
k. a host bus coupled to the host processor;
l. an internal bus;
25 m. a first means for storing and updating elements of the solution vector, the first means coupled to the host bus and the internal bus;
n. a second means for storing elements of the coefficient matrix, the second means coupled to the host and internal buses;
o. a third means for determining successful iterations and preferable
30 updates, the third means coupled to the internal bus and the second means;

p. a fourth means for storing and updating elements of an auxiliary vector,
the fourth means coupled to the host bus, the internal bus, and the second
means.

5 109. The computer system of claim 108, comprising a controller coupled to the
host bus and the internal bus, receiving control signals from the host processor
through the host bus and producing control signals for the internal bus.

110. The computer system of claim 109, wherein the first means contains a
10 memory means for storing elements of the solution vector and a means for adding or
subtracting, and the first means updates elements of the solution vector by adding or
subtracting a step-size parameter.

111. The computer system of claim 108, wherein the second means contains a
15 memory means for storing elements of the coefficient matrix and a means for bit-
shifting, and the second means performs bit-shifts of elements of the coefficient
matrix.

112. The computer system of claims 108, wherein the fourth means contains a
20 memory means for storing elements of the auxiliary vector and a means for adding or
subtracting, and the fourth means updates elements of the auxiliary vector by adding
or subtracting bit-shifted elements of the coefficient matrix.

113. The computer system of claim 108, wherein the fourth means receives
25 initialisation data from the host bus.

114. The computer system of claim 108, wherein the fourth means receives
initialisation data from the host bus, the initialisation data being elements of the
right-side vector of the system of linear equations.

30

115. The computer system of claim 108, wherein the second means receives
elements of the coefficient matrix from the host bus.

116. The computer system of claim 108, wherein the third means determines successful iterations by comparing elements of the auxiliary vector and bit-shifted elements of the coefficient matrix.

5

117. A multiuser receiving method in a data transmission system in which code division multiple access, involving multiuser interference among respective signals, each signal representing a succession of data signals translated into bits and transmitted at a rate of a plurality of chips per bit, spread by a respective spreading
10 code, is applied for detecting a particular data signal, from among a plurality of data signals, said method comprising:

- q. filtering matched with the spreading codes and applied to the received signal to obtain respective output signals;
- r. transforming the matched-filter output signals by solving a system of
15 linear equations of the kind $\mathbf{R}\mathbf{h}=\boldsymbol{\beta}$ where \mathbf{R} is a $N \times N$ cross-correlation matrix of the spreading codes, $\boldsymbol{\beta}$ is a $N \times 1$ vector grouping the matched-filter output signals, \mathbf{h} is the $N \times 1$ solution vector representing the transformed signals, and N is a number of used spreading codes, wherein solving the system of linear equations comprises the steps of:
 - 20 i. representing elements of a solution vector as fixed-point binary words each comprising at least one bit;
 - ii. initialising the solution vector and an auxiliary vector;
 - iii. performing, for each bit representing the binary words, bit-wise iterations comprising the steps of: performing passes
25 through all elements of the solution vector; updating elements of the solution vector in the passes; updating elements of the auxiliary vector in the passes; repeating the passes until a finishing condition is fulfilled;
 - iv. stopping solving the system of linear equations when a
30 stopping condition is fulfilled.
- s. subjecting the transformed signals to obtain estimates of the data signals.

118. A multiuser receiver in a data transmission system in which code division multiple access, involving multiuser interference among respective signals, each signal representing a succession of data signals translated into bits and transmitted at a rate of a plurality of chips per bit, spread by a respective spreading code, is applied
5 for detecting a particular data signal, from among a plurality of data signals, said multiuser receiver comprising:

- t. filters matched with spreading codes contained in the received signals;
- u. a computer system for solving systems of linear equations of the kind $\mathbf{R}\mathbf{h}=\boldsymbol{\beta}$ where \mathbf{R} is a $N \times N$ cross-correlation matrix of the spreading codes, $\boldsymbol{\beta}$
10 is a $N \times 1$ vector grouping the matched-filter output signals, \mathbf{h} is the $N \times 1$ solution vector representing the output signals of the computer system, and N is a number of used spreading codes, wherein the computer system performs a sequence of operations comprising the steps of:
 - i. representing elements of a solution vector as fixed-point
15 binary words each consisting of at least one bit;
 - ii. initialising the solution vector and an auxiliary vector;
 - iii. performing, for each bit representing the binary words, bit-wise iterations comprising the steps of: performing passes through all elements of the solution vector; updating elements
20 of the solution vector in the passes; updating elements of the auxiliary vector in the passes; repeating the passes until a finishing condition is fulfilled;
 - iv. stopping solving the system of linear equations when a stopping condition is fulfilled.
- v. a means for estimating the data signals from the output signals of the
25 computer system.

119. A multiuser receiver according to claim 34, wherein the computer system for solving the system of linear equations comprises;
30 w. a host processor receiving parameter signals representing elements of the cross-correlation matrix of the spreading codes and right-side vector of

- the system of linear equations and transmitting parameter signals representing elements of the solution vector;
- x. a host bus coupled to the host processor;
 - y. an internal bus;
 - 5 z. a first means for storing and updating elements of the solution vector, the first means coupled to the host bus and the internal bus;
 - aa. a second means for storing elements of the cross-correlation matrix, the second means coupled to the host bus and the internal bus;
 - bb. a third means for determining successful iterations and preferable
10 updates, the third means coupled to the internal bus and the second means;
 - cc. a fourth means for updating and storing elements of an auxiliary vector, the fourth means coupled to the host bus, the internal bus, and the second means.
- 15
120. An adaptive filter for receiving a first signal from a first transmission line and a second signal from a second transmission line, said first signal partially leaking from said first transmission line to said second transmission line as an echo, said adaptive filter comprising a filter means coupled to said first transmission line and
20 responsive to said first signal for producing an estimated echo signal determined in accordance with filter coefficients, a coefficient generating means for generating said filter coefficients and subtracting means coupled to said filter means and connected in said second transmission line for subtracting said estimated echo signal from said second signal on said second transmission line so as to cancel said echo signal, said
25 coefficient generating means comprises:
- dd. a first means coupled to said first transmission line and responsive to said first signal for producing a series of autocorrelation coefficients of said first signal;
 - ee. a second means coupled to said first and said second transmission lines
30 and responsive to said first and said second signal for producing a series of cross-correlation coefficients between said first signal and said second signal; and

- 5 ff. a means coupled to said first and said second means for generating said filter coefficients from said autocorrelation and cross-correlation coefficients to deliver said filter coefficients to said filter means, said third means is a computer system for solving a system of linear equations whose coefficient matrix comprises said autocorrelation coefficients and whose right-side vector comprises said cross-correlation coefficients, wherein said computer system for solving the system of linear equations performs a sequence of operations comprising the steps of:
- 10 i. representing elements of a solution vector as fixed-point binary words each consisting of at least one bit;
- ii. initialising the solution vector and an auxiliary vector;
- 15 iii. performing, for each bit representing the binary words, bit-wise iterations comprising the steps of: performing passes through all elements of the solution vector updating elements of the solution vector in the passes; updating elements of the auxiliary vector in the passes; repeating the passes until a finishing condition is fulfilled;
- iv. stopping solving the system of linear equations when a stopping condition is fulfilled.
- 20 121. The adaptive filter of claim 120, wherein the computer system for solving the system of linear equations comprises:
- gg. a host processor receiving parameter signals representing elements of the coefficient matrix and the right-side vector and transmitting parameter signals representing elements of the solution vector;
- 25 hh. a host bus coupled to the host processor;
- ii. an internal bus;
- jj. a first means for storing and updating elements of the solution vector, the first means coupled to the host bus and the internal bus;
- 30 kk. a second means for storing elements of the coefficient matrix, the second means coupled to the host bus and the internal bus;

ll. a third means for determining successful iterations and preferable updates, the third means coupled to the internal bus and the second means;

5 mm. a fourth means for storing and updating elements of an auxiliary vector, the fourth means coupled to the host bus, internal bus, and the second means.

122. The adaptive filter of claim 121, wherein said filter means is a transversal filter.

10